Refine Search

Search Results -

Terms	Documents
L10 and @pd > 20051106	0

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L11

Refine Search

Recall Text | Clear | Interrupt

Search History

DATE: Thursday, June 22, 2006 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=NO; OP=OR			
L11	L10 and @pd > 20051106	0	L11
L10	L9 AND html	0	L10
L9	L8 AND trigger and token and pars$3 and lex$	7	L9
L8	L7 OR L6 OR L4 OR L3 OR L2 OR L1	482	L8
L7	(715/542).ccls.	111	L7
L6	707/534	100	L6
L5	707/534.ccls	0	L5
L4	(717/113).ccls.	118	L4
L3	(717/111).ccls.	57	L3
L2	(717/110).ccls.	107	L2
L1	(717/112).ccls.	25	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
L16 and editor	24

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L17

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Thursday, June 22, 2006 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=PGPB; PLUR=NO; OP=OR			
L17	L16 and editor	24	L17
L16	L15 and cursor	39	L16
L15	L13 or l12	143	L15
L14	l12 and l13	0	L14
L13	trigger and token and pars$3 and lex$	135	L13
L12	(717/112).ccls.	8	L12
DB=USPT; PLUR=NO; OP=OR			
L11	L10 and @pd > 20051106	0	L11
L10	L9 AND html	0	L10
L9	L8 AND trigger and token and pars$3 and lex$	7	L9
L8	L7 OR L6 OR L4 OR L3 OR L2 OR L1	482	L8
L7	(715/542).ccls.	111	L7
L6	707/534	100	L6
L5	707/534.ccls	0	L5

L4	(717/113).ccls.	118	L4
L3	(717/111).ccls.	57	L3
L2	(717/110).ccls.	107	L2
L1	(717/112).ccls.	25	L1

END OF SEARCH HISTORY

Terms used **syntax directed editor buffer cursor**

Found **48,440** of **178,880**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Automatically generating visual syntax-directed editors

Farah Arefi, Charles E. Hughes, David A. Workman

March 1990 **Communications of the ACM**, Volume 33 Issue 3

Publisher: ACM Press

Full text available: pdf(3.22 MB) Additional Information: full citation, references, citings, index terms, review

Keywords: abstract syntax tree schemas, computer-aided software engineering (CASE), plan diagrams, reuse

5 Multifunctional cursor for direct manipulation user interfaces

M. J. Muller

May 1988 **Proceedings of the SIGCHI conference on Human factors in computing systems**

Publisher: ACM Press

Full text available: pdf(626.04 KB) Additional Information: full citation, abstract, references, citings, index terms

The multifunctional cursor (MC) is a technique for representing multiple operations in direct manipulation user interfaces. Icons for each of several simultaneously-available operations are overlaid into the cursor image. The MC improves user interface practice by removing syntactic inconsistencies, by reducing cognitive load, and by providing support for repeated operations.

6 Z - the 95% program editor

Steven R. Wood

June 1981 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN SIGOA symposium on Text manipulation**, Volume 16 Issue 6

Publisher: ACM Press

Full text available: pdf(757.05 KB) Additional Information: full citation, abstract, references, citings, index terms

Recently much attention has been focused on structure-oriented program editors that have specific knowledge about the syntax and semantics of a particular programming language [1, 4, 5, 18]. These editors provide many desirable features for editing programs. However, the user interface is constrained by the syntax and semantics of the target language, and editing operations that are simple in a text editor can be quite complicated in a structure-oriented editor. In addition, the user has an ...

7 The Pan language-based editing system for integrated development

Robert A. Ballance, Susan L. Graham, Michael L. Van De Vanter

October 1990 **ACM SIGSOFT Software Engineering Notes , Proceedings of the fourth ACM SIGSOFT symposium on Software development environments SDE 4**, Volume 15 Issue 6

Publisher: ACM Press

Full text available: pdf(2.11 MB) Additional Information: full citation, abstract, references, citings, index terms

Powerful editing systems for developing complex software documents are difficult to engineer. Besides requiring efficient incremental algorithms and complex data structures, such editors must integrate smoothly with the other tools in the environment, maintain a sharable database of information concerning the documents being edited, accommodate flexible editing styles, provide a consistent, coherent, and empowering user interface, and support individual variations and project-wide configura ...

8 On-line Text Editing: A Survey

Andries van Dam, David E. Rice

September 1971 **ACM Computing Surveys (CSUR)**, Volume 3 Issue 3

Publisher: ACM Press

Full text available: pdf(1.91 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper is a survey of current methods for the on-line creation and editing of computer programs and of ordinary manuscripts text. The characteristics of on-line editing systems are examined and examples of various implementations are described in three categories: program editors, text editors, and terminals with local editing facilities.

9 The JADE interpreter: a RISC interpreter for syntax directed editing

C. F. Clark

July 1987 **ACM SIGPLAN Notices , Papers of the Symposium on Interpreters and interpretive techniques SIGPLAN '87**, Volume 22 Issue 7

Publisher: ACM Press

Full text available: pdf(425.38 KB) Additional Information: full citation, abstract, index terms

This paper describes key features of an interpreter for a language-based editor. The interpreter unites in a RISC framework features which have been used in other domains. The paper examines each feature's integration into the RISC framework.

10 Techniques for improving language-based editors

J. R. Horgan, D. J. Moore

April 1984 **ACM SIGSOFT Software Engineering Notes , ACM SIGPLAN Notices , Proceedings of the first ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments SDE 1**, Volume 9 , 19 Issue 3 , 5

Publisher: ACM Press

Full text available: pdf(551.88 KB) Additional Information: full citation, abstract, references, citings, index terms

Syned is a working language-based editor which includes a complete parser for editing the C language. The design ideas which permit parsing in Syned also result in the solution of several important language-based editor problems. We describe these ideas in sufficient detail to make them accessible to others.

11 A PDP-8 emulator program

Brian J. Shelburne

March 2002 **Journal on Educational Resources in Computing (JERIC)**, Volume 2 Issue 1

Publisher: ACM Press

Full text available: pdf(270.03 KB) Additional Information: full citation, abstract, references, citings, index terms

The clean, simple, and elegant architecture of the classic PDP-8 makes it an ideal candidate for studying concepts in computer organization. The PDP-8 emulator program allows a user to write, edit, assemble, debug, trace, and execute PDP-8 machine code and PDP-8 assembler language programs. With it, the user can obtain a feel for the PDP-8. The PDP-8 emulator program includes a simple built-in text editor which is used to write and edit PDP-8 assembler language programs, an assembler to translat ...

Keywords: Computer architecture simulator, education

12 Efficient abstractions for the implementation of structured editors

Robert Hood

June 1985 **ACM SIGPLAN Notices , ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 85 symposium on Language issues in programming environments**, Volume 20 , 18 Issue 7 , 6

Publisher: ACM Press

Full text available: pdf(539.51 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper investigates the use of abstract recursive data structures and operations in the implementation of a structured program editor. The value-oriented semantics of the proposed constructs simplify the implementation of important features such as version control and an unbounded undo operation. Since the constructs can be implemented efficiently, their use in the structured program editor does not significantly affect its performance.

13 The Poe language-based editor project

C. N. Fischer, Gregory F. Johnson, Jon Mauney, Anil Pal, Daniel L. Stock
April 1984 **ACM SIGSOFT Software Engineering Notes , ACM SIGPLAN Notices , Proceedings of the first ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments SDE 1**, Volume 9 , 19 Issue 3 , 5
Publisher: ACM Press

Full text available: pdf(708.85 KB) Additional Information: full citation, abstract, references, citings, index terms

Editor Allan Poe (Pascal Oriented Editor) is a full-screen language-based editor (LBE) that knows the syntactic and semantic rules of Pascal. It is the first step in development of a comprehensive Pascal program development environment. Poe's design began in 1979; version 1 is currently operational on Vax 11s under Berkeley Unix and on HP 9800-series personal workstations. Poe is written in Pascal, and is designed to be readily transportable to new machines. An editor-generating ...

14 Draft Proposed: American National Standard—Graphical Kernel System

Technical Committee X3H3 - Computer Graphics
February 1984 **ACM SIGGRAPH Computer Graphics**, Volume 18 Issue SI
Publisher: ACM Press
Full text available: pdf(16.07 MB) Additional Information: full citation

15 Parallel and distributed incremental attribute evaluation algorithms for multiuser software development environments

Gail E. Kaiser, Simon M. Kaplan
January 1993 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 2 Issue 1
Publisher: ACM Press

Full text available: pdf(3.09 MB) Additional Information: full citation, abstract, references, citings, index terms

The problem of change propagation in multiuser software development environments distributed across a local-area network is addressed. The program is modeled as an attributed parse tree segmented among multiple user processes and changes are modeled as subtree replacements requested asynchronously by individual users. Change propagation is then implemented using decentralized incremental evaluation of an attribute grammar that defines the static semantic properties of the p ...

Keywords: attribute grammar, change propagation, distributed, incremental algorithm, parallel, reliability

16 Human-computer interface development: concepts and systems for its management

H. Rex Hartson, Deborah Hix
March 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 1

Publisher: ACM Press

Full text available: pdf(7.97 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Human-computer interface management, from a computer science viewpoint, focuses on the process of developing quality human-computer interfaces, including their representation, design, implementation, execution, evaluation, and maintenance. This survey presents important concepts of interface management: dialogue independence, structural modeling, representation, interactive tools, rapid prototyping, development methodologies, and control structures. *Dialogue independence* is th ...

17 A structural view of the Cedar programming environment

Daniel C. Swinehart, Polle T. Zellweger, Richard J. Beach, Robert B. Hagmann
August 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 4
Publisher: ACM Press

Full text available: pdf(6.32 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents an overview of the Cedar programming environment, focusing on its overall structure—that is, the major components of Cedar and the way they are organized. Cedar supports the development of programs written in a single programming language, also called Cedar. Its primary purpose is to increase the productivity of programmers whose activities include experimental programming and the development of prototype software systems for a high-performance personal computer. T ...

18 Incremental dynamic semantics for language-based programming environments

G. E. Kaiser
April 1989 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 11 Issue 2
Publisher: ACM Press

Full text available: pdf(1.99 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Attribute grammars are a formal notation for expressing the static semantics of programming languages—those properties that can be derived from inspection of the program text. Attribute grammars have become popular as a mechanism for generating language-based programming environments that incrementally perform symbol resolution, type checking, code generation, and derivation of other static semantic properties as the program is modified. However, attribute grammars are not suitable fo ...

19 Building integrated software development environments. Part I: tool specification

G. Engels, C. Lewerentz, M. Nagl, W. Schäfer, A. Schürr
April 1992 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 1 Issue 2
Publisher: ACM Press

Full text available: pdf(2.19 MB) Additional Information: full citation, abstract, references, citings, index terms

The conceptual modeling approach of the IPSEN (Integrated Project Support Environment) project for building highly integrated environments is based on using attributed graphs to model and implement arbitrary object structures, in particular all kinds of software documents and their relationships. A language based on graph grammars, called PROGRESS (PROgrammed Graph REwriting SyStems), and a suitable method for the application of this language, called graph grammar engineering, have been dev ...

Keywords: attribute grammars, attributed graphs, environment generators, graph grammars

20 How interface design determines Who has difficulty learning to use a text editor

Louis M. Gomez, Dennis E. Egan, Evangeline A. Wheeler, Dhiraj K. Sharma, Aleta M. Gruchacz

December 1983 **Proceedings of the SIGCHI conference on Human Factors in Computing Systems**

Publisher: ACM Press

Full text available: pdf(490.27 KB) Additional Information: full citation, abstract, references, citings, index terms

In previous studies two background characteristics of computer novices were consistently correlated with their success in learning to use a line-based computer text editor. Older people and those who scored low on a standard test of Spatial Memory had more difficulty than younger people and those with higher Spatial Memory test scores. In the present study, we observed computer novices as they learned to use a screen-based editor, which presumably reduced spatial memory load. Contr ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next